



TITLE:

Branch-length とtree-length(計算機科学の理論とその応用)

AUTHOR(S):

梅澤, 香織; 山崎, 浩一

CITATION:

梅澤, 香織 ...[et al]. Branch-length とtree-length(計算機科学の理論とその応用). 数理解析研究所講究録 2007, 1554: 230-237

ISSUE DATE:

2007-05

URL:

<http://hdl.handle.net/2433/80948>

RIGHT:

Branch-length と tree-length

群馬大学大学院 工学研究科 梅澤 香織 (K.Umezawa)

群馬大学工学部 情報工学科 山崎 浩一 (K.Yamazaki)

Computer Science, Gunma University

概要

Branch-decomposition 上のパラメータ branch-length を定義し, branch-length が tree-decomposition 上で定義される tree-length と一致することを示す.

キーワード : tree-decomposition, tree-length, branch-decomposition, branch-length

1 はじめに

Branch-width と tree-width は, 古くから盛んに研究されているグラフパラメータである. その理由の 1 つに一般のグラフ上では NP 完全な問題も branch-width または tree-width を制限することで多項式時間で解けることが挙げられる [1, 8].

Branch-width は branch-decomposition 上で定義され, tree-width は tree-decomposition 上で定義される. Branch-width に着目したアルゴリズムの多くは branch-decomposition を用いており, それらの計算量は branch-width に深く依存している. 同様に tree-width に着目したアルゴリズムの多くは tree-decomposition を用い, それらの計算量は tree-width に深く依存している. Branch-decomposition はハイパーグラフやマトロイドへの一般化が自然に行えるという利点を持つ. 一方 tree-decomposition

は直感的にわかりやすく, k -tree, 弦グラフ, 極大クリークサイズ, 部分木による交差表現などは tree-decomposition と深い関わりがある. また branch-width bw と tree-width tw の間には $\max\{bw, 2\} \leq tw + 1 \leq \max\{\frac{3}{2}bw, 2\}$ という関係があることが知られている [7].

Tree-decomposition 上で定義される別のパラメータの 1 つに tree-length がある. Tree-length は弦グラフにどれだけ近いかという指標にもなるパラメータであり, tree-length が制限されたグラフもよい性質を持つ. 例えば tree-length が抑えられたグラフは辺数の少ない spanner を持つことが [9] によって証明されている. あるグラフの spanner とは, そのグラフの部分グラフで, かつ 2 頂点間の距離が元のグラフでの距離とさほど変わらないものをいう. 多くの分散アルゴリズムの計算量は spanner の辺数に依存しているので [4], 辺数の少ない spanner を見つけることはこれらのアルゴリズムの計算量を抑えることになる.

Tree-width に対応する branch-width のような, tree-length に対応するグラフパラメータは研究されていなかった. そこで本論文では tree-length に対応する, branch-decomposition 上のパラメータ branch-length について考え, tree-length との関係について考察する. Tree-width は branch-width の 1.5 倍程度で抑えられるため, tree-length も branch-length の定数倍で抑えら

れることは自然に想像できる。では実際にどのくらいの開きがあるだろうか。本論文では branch-width が 2 以上のとき, tree-length と branch-length が一致することを証明する。

2 準備

グラフ G は連結であるとする。 $V(G)$ のある部分集合を V_X としたとき, V_X の直径 $\max_{u,v \in V_X} d_G(u,v)$ を $\text{diam}_G(V_X)$ と表記する。

定義 1. [6] グラフ G の tree-decomposition $\mathcal{T} = (T, \chi)$ とは, 以下を満たす木 T と $V(G)$ の部分集合の族から成る, T の頂点に対応する集合 (一般に *bag* と呼ぶがここでは *t-node* と呼ぶことにする) の集合 χ の 2 項組とする。

1. $\cup_{X \in \chi} X = V(G)$.
2. G の任意の辺 $\{u, v\}$ に対し, u と v を含む *t-node* X が存在する。
3. ある *t-node* X, Y, Z があるとき, Y が T 上のパス $X - Z$ に含まれるなら $X \cap Z \subseteq Y$ を満たす。

また tree-decomposition の木 T の辺を *t-edge* と呼ぶことにする。

定義の 3 つ目の条件は, “ G の任意の頂点 v に対し, T から v を含む *t-node* を誘導すれば木であること”と置き換えてもよい。例えば, 図 1 のようなグラフ G があるとき, 図 2 は G の tree-decomposition であり, どの頂点 $v \in V(G)$ を含む *t-node* で誘導しても木であることがわかる。

G のある tree-decomposition $\mathcal{T} = (T, \chi)$ があるとき, \mathcal{T} の *width* $\max_{X_i \in \chi} |X_i| - 1$ を $\text{width}(\mathcal{T})$, \mathcal{T} の *length* $\max_{X_i \in \chi} \text{diam}_G(X_i)$ を $\text{length}(\mathcal{T})$ と表記する。さらに G の *tree-width* $\text{tw}(G)$, G の *tree-length* $\text{tl}(G)$ とは G の

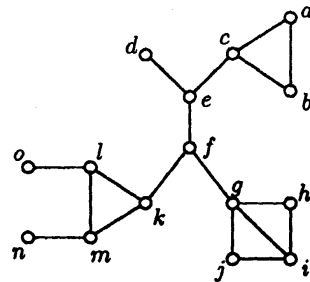


図 1: グラフ G

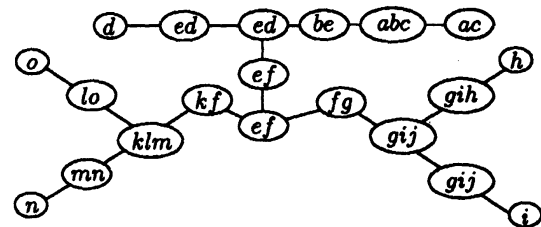


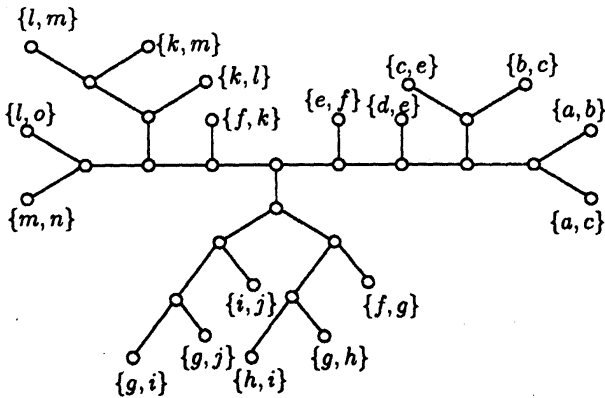
図 2: G のある tree-decomposition

全ての tree-decomposition \mathcal{T} においてそれぞれ $\min_{\mathcal{T}} \text{width}(\mathcal{T})$, $\min_{\mathcal{T}} \text{length}(\mathcal{T})$ とする。

定義 2. [7] G の branch-decomposition $\mathcal{B} = (T, \mu)$ とは, 次数 1 または 3 の木 T と, G の辺集合から T の葉集合への全単射 μ の 2 項組とする。Branch-decomposition の木 T の頂点と辺をそれぞれ *b-node*, *b-edge* と呼ぶことにする。

グラフ G のある branch-decomposition を図 3 に示す。

Branch-decomposition $\mathcal{B} = (T, \mu)$ の各 *b-edge* $\{X, Y\}$ に対し, $\text{mid}(\{X, Y\})$ を定義する。 T の *b-edge* $\{X, Y\}$ を取り除くことで分けられる 2 つの部分木を T_1, T_2 とする。 T_1 に含まれる葉から μ^{-1} で写像される G の辺の両端点の集合を V_1 とする。同様に T_2 に含まれる葉から写像される辺の両端点の集合を V_2 とする。 $\text{mid}(\{X, Y\}) = V_1 \cap V_2$ と定義する。 $\mathcal{B} = (T, \mu)$ の *width* を $\max_{\{X, Y\} \in E(T)} |\text{mid}(\{X, Y\})|$ とし, $\text{width}(\mathcal{B})$

図 3: グラフ G のある branch-decomposition

と表記する。 G の branch-width とは、 G の全ての branch-decomposition B において $\text{min}_B \text{width}(B)$ とし、 $\text{bw}(G)$ と表記する。

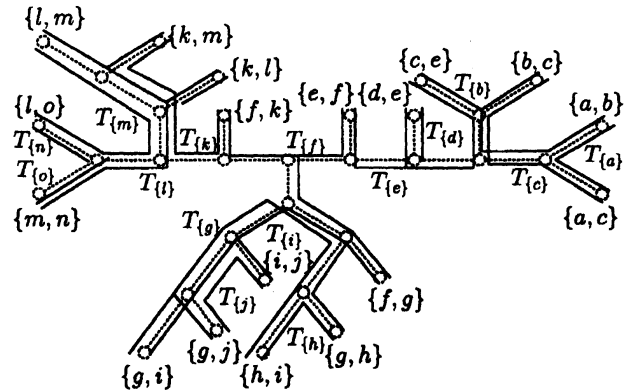
定義 3. G のある branch-decomposition (T, μ) とある頂点 $x \in V(G)$ があるとき、 x を端点とする G の辺を $\{x, y_1\}, \{x, y_2\}, \dots, \{x, y_l\}$ とし、 $b\text{-node } \mu(\{x, y_1\}), \mu(\{x, y_2\}), \dots, \mu(\{x, y_l\})$ を葉として持つ T の部分木を $T_{\{x\}}$ と表記する。 T のある $b\text{-node } X$ に対し、 X を含む部分木 $T_{\{x_1\}}, T_{\{x_2\}}, \dots, T_{\{x_m\}}$ があるとき、 $b\text{-node } X$ に対応する頂点集合 $V_{[X]}$ を $V_{[X]} := \{x_1, x_2, \dots, x_m\}$ と定義する。

例えば図 3 の branch-decomposition は $T_{\{x\}}$, $x \in V(G)$ を用いれば図 4 のように表現できる。

3 Branch-length

Branch-decomposition 上の新たなパラメータ branch-length を定義し、 branch-width が 2 以上のとき branch-length と tree-length が一致することを示す。

定義 4. G のある Branch-decomposition を $B = (T, \mu)$ としたとき、 $B = (T, \mu)$ の length

図 4: 図 3 の $T_{\{x\}}$, $x \in V(G)$ での表現

を $\max_{\{X, Y\} \in E(T)} \text{diam}_G(\text{mid}(\{X, Y\}))$ と定義し、 length(B) と表記する。 G の branch-length $\text{bl}(G)$ とは、 G の全ての branch-decomposition B において $\text{min}_B \text{length}(B)$ とする。

Branch-width が 1 のグラフは、 $\text{width}(B) = 1$ を満たす branch-decomposition B が存在する。 このとき B の任意の $b\text{-edge } \{X, Y\}$ において $|\text{mid}(\{X, Y\})| = 1$ であり、 直径は 0 であるから branch-length は 0 である。 一方 tree-decomposition には辺の両端点を含む $t\text{-node}$ が存在するので tree-length は 1 以上である。 従って本論文では、 branch-width が 2 以上の場合を考える。

Branch-length が tree-length と一致することを証明するために次の 2 つを示す;

1. G の branch-decomposition から length を増やすことなく G の tree-decomposition が作れること (補題 2),
2. G の tree-decomposition から length を増やすことなく G の branch-decomposition が作れること (補題 4).

補題 1. G の branch-decomposition (T, μ) があるとき、 T のある $b\text{-edge } \{X, Y\}$ に対し

$\text{mid}(\{X, Y\})$ に含まれる G の頂点を v とすると, v は $V_{[X]}$ と $V_{[Y]}$ に含まれる.

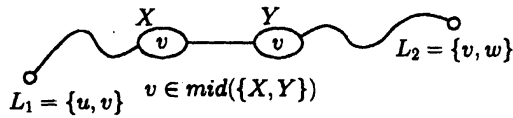


図 5: $v \in \text{mid}(\{X, Y\})$

証明. T から $b\text{-edge}\{X, Y\}$ を取り除くことでできる 2 つの部分木のうち, X を含む方の部分木を T_1 , Y を含む方の部分木を T_2 とする. $\text{mid}(\{X, Y\})$ が v を含むなら定義より T_1 に含まれ $L_1 = \mu^{-1}(\{u, v\})$, $\{u, v\} \in E(G)$ を満たす $b\text{-node}$ L_1 と, T_2 に含まれ $L_2 = \mu^{-1}(\{v, w\})$, $\{v, w\} \in E(G)$ を満たす $b\text{-node}$ L_2 が存在する (図 5 参照). $T_{\{v\}}$ は連結であるから, $T_{\{v\}}$ は X と Y を含む. よって v は $V_{[X]}$ と $V_{[Y]}$ に含まれる. \square

補題 2. 任意のグラフ G に対し, $bl(G) \leq k$ であるならば $tl(G) \leq k$ を満たす.

証明. G の branch-length を δ とし, length が δ と等しい G の branch-decomposition を (T_b, μ) とする. (T_b, μ) から以下のようにしてできる (T_t, χ) は, G の tree-decomposition であり length は高々 k であることを示す.

$T_t := T_b$ とする. $T_t = T_b$ なので T_t の頂点と辺を $t\text{-node}$, $t\text{-edge}$ と呼ぶことにする. T_t の各 $t\text{-node}$ に次のように $V(G)$ の部分集合を割り当てる. T_t の各葉 L には, その葉に写像される G の辺の端点から成る頂点集合を割り当てる. 言い換えると T_b の各葉 L に対し, $\mu^{-1}(L) = \{u, v\}$ なる辺 $\{u, v\} \in E(G)$ があるとき, $L := \{u, v\}$ とする. T_t の次数 3 の $t\text{-node}$ X に対しては, X に接続する $t\text{-edge}$ を $\{X, Y_1\}, \{X, Y_2\}, \{X, Y_3\}$ とすると, 図 6 のように $X := (\text{mid}(\{X, Y_1\}) \cap \text{mid}(\{X, Y_2\})) \cup$

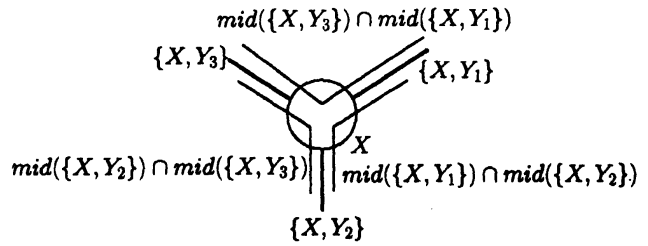


図 6: $b\text{-node } X = (\text{mid}(\{X, Y_1\}) \cap \text{mid}(\{X, Y_2\})) \cup (\text{mid}(\{X, Y_2\}) \cap \text{mid}(\{X, Y_3\})) \cup (\text{mid}(\{X, Y_3\}) \cap \text{mid}(\{X, Y_1\}))$

$(\text{mid}(\{X, Y_2\}) \cap \text{mid}(\{X, Y_3\})) \cup (\text{mid}(\{X, Y_3\}) \cap \text{mid}(\{X, Y_1\}))$ とする. 以上のように $t\text{-node}$ に割り当てた頂点集合の族を χ とする.

まず (T_t, χ) が tree-decomposition であることを証明する. (T_b, μ) は branch-decomposition であるから T_t は木である. G は孤立点を持たないから, G のどの頂点にもその頂点を端点とする辺が存在する. Branch-decomposition の定義より G のある辺を $\{u, v\}$ とすると $L = \mu(\{u, v\})$ なる $t\text{-node}$ L が存在する. (T_t, χ) の作り方から u と v は L に含まれる. よって G の任意の頂点において, それを含む集合が χ に存在し, さらに G の任意の辺 $\{u, v\}$ において, u と v を含む $t\text{-node}$ が存在する.

G の任意の頂点 x において, x を含む $t\text{-node}$ で T_t を誘導した部分グラフは木であることを示す. (T_b, μ) は branch-decomposition であるから $T_{b\{x\}}$ は木である. x を含む $t\text{-node}$ で T_t を誘導した部分グラフと $T_{b\{x\}}$ が一致することを示せばよい.

X を頂点 x を含むある $t\text{-node}$ とし, X が $T_{b\{x\}}$ に含まれることを示す. X が葉であるなら X には x を端点とする G の辺が写像されているので $T_{b\{x\}}$ は X を含む. X が葉でないとき, X に接続する 3 辺を $\{X, Y_1\}, \{X, Y_2\}, \{X, Y_3\}$ とする. $x \in X$ であることから, 一般性を失うことなく x は $\text{mid}(\{X, Y_1\}) \cap \text{mid}(\{X, Y_2\})$ に含まれてい

たと仮定できる。このとき $x \in \text{mid}(\{X, Y_1\})$ と補題 1 より, x は $V_{[X]}$ に含まれる。すなわち X は $T_{b\{x\}}$ に含まれる。

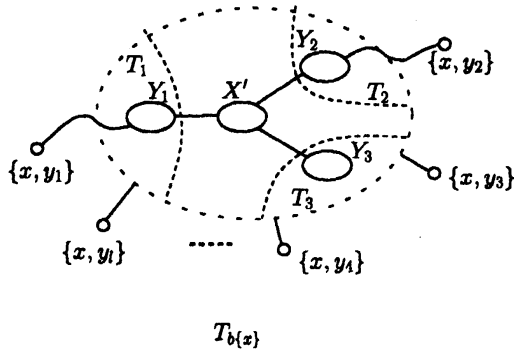


図 7: $X \in T_{b\{x\}}$

X' を, $T_{b\{x\}}$ に含まれる T_i の t -node とし, X' は x を含むことを示す。 X' が葉であるなら作り方から X' は x を含む。 X' が葉でないとき, X' に接続する 3 辺を $\{X', Y_1\}$, $\{X', Y_2\}$, $\{X', Y_3\}$ とし, X' を取り除くことで分けられる 3 つの部分木のうち $\{X', Y_1\}$ に接続するものを T_1 , 同様に $\{X', Y_2\}$, $\{X', Y_3\}$ に接続するものを T_2, T_3 とする。図 7 は $T_{b\{x\}}$ を表している。 $T_{b\{x\}}$ の葉は x が写像された t -node $\mu(\{x, y_1\}), \mu(\{x, y_2\}), \dots, \mu(\{x, y_l\})$ から成る。 $X' \in V(T_{b\{x\}})$ より T_1, T_2, T_3 の少なくとも 2 つには x が写像されている。 T_1 と T_2 に x が写像されているとすると, $\text{mid}(\{X, Y_1\})$ は x を含み, また $\text{mid}(\{X, Y_2\})$ も x を含む。従って作り方から X' は x を含む。

以上より x を含む t -node で誘導した部分グラフ T_x は木であるので (T_b, χ) は tree-decomposition である。

次に (T_i, χ) が $\text{length}((T_i, \chi)) \leq k$ を満たすことを示す。 T_i のある葉 L は隣接した 2 頂点から成るので L の length すなわち $\text{diam}_G(L)$ は 1 である。 T_i の葉ではない t -node X は上の表記を用いると, $X = (\text{mid}(\{X, Y_1\}) \cap$

$\text{mid}(\{X, Y_2\})) \cup (\text{mid}(\{X, Y_2\}) \cap \text{mid}(\{X, Y_3\})) \cup (\text{mid}(\{X, Y_3\}) \cap \text{mid}(\{X, Y_1\}))$ である。従ってこの t -node X に含まれるどの 2 頂点も $\text{mid}(\{X, Y_1\})$, $\text{mid}(\{X, Y_2\})$, $\text{mid}(\{X, Y_3\})$ の少なくともどれか 1 つに (両方) 含まれる。仮定より $\text{length}((T_b, \mu))$ は k 以下であるので $(1 \leq) \text{length}(X) \leq k$ を満たす。どの t -node X_i においても $\text{diam}_G(X_i)$ は k 以下であるので T_i の length も高々 k であることが示せた。 \square

定義 5. G の tree-decomposition (T, χ) があり, G の各辺 $\{u, v\}$ において, G の頂点 u と v を含む t -node X が 1 つ定められているとき, X は辺 $\{u, v\}$ に assignment されているという。

Tree-decomposition の定義より, G の各辺に対し, 両端点を含む t -node が存在するのでその辺に assignment されている t -node を 1 つ定めることができる。

定義 6. G の tree-decomposition (T, χ) が, G の各辺 $\{u, v\}$ において assignment する t -node が定められているとする。任意の t -edge $\{X, Y\}$ に対し, $\text{mid}'(\{X, Y\})$ を定義する。 $\{X, Y\}$ を取り除いてできる T の部分木を T_1, T_2 とする。 T_1 の t -node に assignment する全ての辺の両端点の集合を V_1 とする。同様に T_2 の t -node に assignment する全ての辺の両端点の集合を V_2 とする。 $\text{mid}'(\{X, Y\}) := V_1 \cap V_2$ とする。

補題 3. G のある tree-decomposition (T, χ) があり, G の各辺において assignment する t -node が定まっているとする。 T のある t -edge $\{X, Y\} \in E(T)$ に対し $\text{mid}'(\{X, Y\})$ に含まれる G の頂点を v とすると, v は t -node X と t -node Y に含まれる。

証明. T から t -edge $\{X, Y\}$ を取り除くことでできる 2 つの部分木のうち, X を含む方の部分木を T_1 , Y を含む方の部分木を T_2 とする。

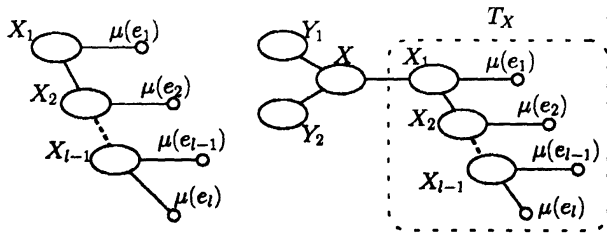


図 8: T_X , X の次数が 2 で assignment されているとき

$mid'(\{X, Y\})$ に含まれる頂点を v とすると, v は T_1 の t -node に assignment するある辺の端点であり, T_2 の t -node に assignment する別の辺の端点でもある. v を端点とする辺が assignment する T_1 のある t -node を X' , T_2 のある t -node を Y' とする. 定義 5 より, v は X', Y' に含まれる. また, X と Y は X' から Y' へのパスに含まれる. 従って tree-decomposition の定義より, t -node X と t -node Y は v を含む. \square

補題 4. 任意のグラフ G に対し, $tl(G) \leq k$ であるならば $bl(G) \leq k$ を満たす.

証明. δ を G の tree-length とし, length が δ の G の tree-decomposition を (T_t, χ) とする. (T_t, χ) から G の branch-decomposition (T_b, μ) への変換を与え, 結果得られる (T_b, μ) が $length((T_b, \mu)) \leq k$ を満たすことを示す. 変換は次の 3 ステップからなる.

1. G の各辺 $\{u, v\} \in E(G)$ において, χ の中から u と v を含む t -node を 1 つ assignment する.
2. 葉は全て G の辺に assignment された t -node になるように T_t を縮約し, 結果を (T'_t, χ') とする.
3. (T'_t, χ') の各 t -node X に対し以下の操作を行ったものを (T_b, μ) とする.

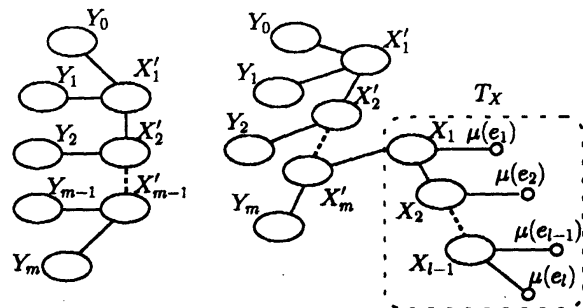


図 9: X の次数が 4 以上で assignment されていないとき, X の次数が 3 以上で assignment されているとき

G の各辺 $\{u, v\}$ において $\mu(\{u, v\})$ なる頂点を用意する. ある t -node X に assignment する G の辺が l ($l \geq 2$) 本あるとき, それらの辺を e_1, e_2, \dots, e_l とする. X の $l-1$ 個のコピーから成るパス X_1, \dots, X_{l-1} に t -edge $\{X_i, \mu(e_i)\}$, $1 \leq i \leq l-1$ と t -edge $\{X_{l-1}, \mu(e_l)\}$ を加えた木を T_X と呼ぶことにする (図 8 参照). このとき X_1 を, T_X の根とする. $l=1$ のときは X を T_X とする.

X の次数によって次のような操作を行う.

- (a) X の次数が 1 のとき
(T'_t, χ') の作り方から X は assignment されている. X を, X を assignment する辺の木 T_X と置き換える. このとき X と隣接していた t -node と T_X の根を辺で結ぶ.
- (b) X の次数が 2 のとき, X が assignment されていないならば X を縮約する. すなわち, X に隣接する t -node を Y と Z としたとき, X を削除し, Y と Z 間に辺を加える. X が assignment されているならば, T_X を加えて T_X の根と X を辺で結ぶ.
- (c) X の次数が 3 以上のとき,
 X に隣接する t -node を Y_0, Y_1, \dots, Y_m

とする。図9に次数3以上のときの変換を与える。

X が assignment されていないとき, X の次数が4以上なら X を, X のコピーから成るパス X'_1, \dots, X'_{m-1} に置き換える. t -edge $\{X'_i, Y_i\}, 1 \leq i \leq m-1$ と t -edge $\{X'_1, Y_0\}, \{X'_{m-1}, Y_m\}$ を加える。

X が assignment されているなら X を, X のコピーから成るパス X'_1, \dots, X'_m と置き換え, t -edge $\{X'_i, Y_i\}, 1 \leq i \leq m$ と $\{X'_1, Y_0\}$ を加える. さらに T_X を加え, T_X の根と X'_m を辺で結ぶ。

(T_b, μ) は branch-decomposition である。なぜなら (T_b, μ) の t -node は次数1または3であり, G の各辺は全単射 μ によって T_b の葉に写像されているからである。これ以降は $\mu(e_i), \{X_i, X'_i\}$ 等の頂点, 辺も b -node, b -edge と呼ぶ。

(T_t, χ) の tree-length は高々 k である。従って $\text{length}((T_b, \mu)) \leq k$ であることは, 全ての b -edge $\{X, Y\} \in E(T_b)$ に対し $\text{mid}(\{X, Y\})$ を含む tree-decomposition (T_t, χ) の t -node が存在することを示せばよい。 T'_t の各 b -edge と3の操作で加えられる b -edge に分けて考える。

ステップ2で得られる (T'_t, χ') のある b -edge を $\{X, Y\}$ とする。 $\text{mid}'(\{X, Y\})$ と, 得られた branch-decomposition (T_b, μ) の $\text{mid}(\{X, Y\})$ は一致するので $\text{mid}'(\{X, Y\})$ について証明すればよい。補題2より, $\text{mid}'(\{X, Y\})$ に含まれる頂点は (T_t, χ) の t -node X に含まれる。またステップ3の操作で加えられる t -edge は, (T_t, χ) のある t -node を一方または両方の端点としている。また (T_b, μ) のある b -node X において $V_{[X]}$ は, (T_t, χ) の対応する t -node と等しい。従って補題1より $\text{mid}(\{X, Y\})$ は $V_{[X]}$ に含まれることから, $\text{mid}(\{X, Y\})$ を含む (T_t, χ) に t -node が存在することが示せた。

以上より, $bl(G) \leq k$ である。 \square

補題2, 補題4より,

定理1. $bw(G) \geq 2$ を満たす任意のグラフ G に対し, $tl(G) \leq k \Leftrightarrow bl(G) \leq k$.

4 まとめ

本論文では, branch-width が2以上なら tree-length と branch-length が一致することを証明した。

今後の課題としては, tree-length を branch-decomposition 上で求めるアルゴリズムの設計が挙げられる。既存の tree-length を求める近似アルゴリズムには, 例えば辞書式アルゴリズム LexM [2] や, 距離構造を用いた BFS-Layering [3] 等がある。これらのアルゴリズムはいずれも3倍近似アルゴリズムである。他には2倍近似であると予想されるアルゴリズム Disk-Tree [3] があるが, それよりよい近似アルゴリズムはまだ知られていない。我々は tree-length と branch-length が一致することを示したことにより, tree-decomposition と branch-decomposition のよい性質を互いに利用できるようになった。そこで branch-decomposition のテクニックや道具などを用いて tree-length の近似精度を向上させることが今後の課題である。

参考文献

- [1] Hans L. Bodlaender. A tourist guide through treewidth. *Acta Cybernetica*, 11:1-21, 1993.
- [2] R. Endre Tarjan Donald Rose and George S. Lueker. Algorithmic aspects of vertex elimination on graphs. *SIAM Journal on Computing*, 5:266-283, 1976.

- [3] Yon Dourisboure and Cyril Gavoille. Tree-Decompositions with bags of small diameter. *Discrete Mathematics*, 2005, To appear, extended abstract published in EU-ROCOMB '03. Appears also as RR-1326-01(2004).
- [4] David Peleg and Jeffrey D. Ullman. An optimal synchronizer for the hypercube. *SIAM Journal on Computing*, 18:740–747, 1989.
- [5] Neil Robertson and Paul D. Seymour. Graph minors.I.Excluding a forest. *J. Comb. Theory, Series B*:35:39–61, 1983.
- [6] Neil Robertson and Paul D. Seymour. Graph minors. II. Algorithmic aspects of tree-width. *Journal of Algorithms*, 7:309–322, 1986.
- [7] Neil Robertson and Paul D. Seymour. Graph minors. X. obstructions to tree-decomposition. *J. Combin. Theory Series. B*, 52:153–190, 1991.
- [8] Jan Arne Telle and Andrzej Proskurowski. Algorithms for vertex partitioning problems on partial k -trees. *SIAM Journal on Discrete Mathematics*, 10(4):529–550, 1997.
- [9] Cyril Gavoille Yon Dourisboure, Feodor F. Dragon and Chenyu Yan. Improved spanners for bounded tree-length. *SIROCCO'04*, 2004.